Particle Swarm Optimization (PSO) based feature selection for DDOS attack detection in IOT

S.B Gopal[1*], D. Nanthiya[2], Muthuraman Saminathan[3]
[1]*Department of CSD,* [2]*Department of CT-UG, Kongu Engineering College,Tamil Nadu, India*
[3]*Schlumberger Technology Corporation, Houston, Texas, USA*
**Corresponding Author Email: s.b.gopalece@gmail.com*
Received: 27.09.2025 *Revised: 05.10.2025* *Accepted: 10.10.2025* *Published: 12.10.2025*

Abstract: The Internet of Things (IoT) has experienced rapid expansion, which has resulted in an increased vulnerability to Distributed Denial of Service (DDoS) assaults. These attacks pose significant dangers to environments that are struggling with limited resources. When confronted with high-dimensional datasets, traditional intrusion detection systems (IDS) sometimes experience difficulties, which results in inefficient detection and a rise in the number of false alarms. In order to overcome this issue, a feature selection strategy that based on Particle Swarm Optimization (PSO) is proposed and paired with a deep learning model for efficient DDoS detection in Internet of Things (IoT) networks. A rigorous preparation of benchmark datasets, such as BoT-IoT, SDN-IoT, and KDDCUP1999, provides the foundation for the methodology. This preprocessing ensures that the data is both consistent and accurate. After that, PSO is utilized to choose the features that are most pertinent, so dramatically lowering the dimensionality of the data while maintaining the essential attack-related characteristics. In order to improve learning stability and reduce overfitting, an Artificial Neural Network (ANN) is trained using the optimized dataset. The training process includes batch normalization, dropout regularization, and early stopping. Achieving up to 96% accuracy with reduced false positives and greater generalization, the experimental results reveal significant gains in accuracy, precision, recall, and F1-score. The results demonstrate that the use of PSO-based feature selection improves detection efficiency, which makes the system appropriate for Internet of Things contexts that have limited computational resources.

Keywords: Internet of Things; Distributed Denial of Service; ANN; PSO

1. Introduction

An innovative technological breakthrough, the Internet of Things (IoT) links billions of devices worldwide to collect, distribute, and analyze data in real-time. Smart homes, wearable technology, industrial automation, and healthcare monitoring are just a few examples of how the Internet of Things (IoT) has transformed sectors through increased productivity, streamlined workflows, and improved user experience. However, there are significant security issues associated with the mass deployment of IoT devices. Due to their typically inadequate security measures and low processing and memory capacities, the majority of IoT devices are easy targets for hackers. There is a greater chanceof security breaches when more IoT devices are connected tovital infrastructures, which puts users and businesses at danger. Distributed Denial of Service (DDoS) assaults are one of these risks that should be taken very seriously. An intentional attempt to disrupt normal network, server, orservice traffic by flooding the target with excessive traffic is known as a denial-of-service assault (DDoS). DDoS assaultstake use of the huge quantity of IoT devices, employing themas a botnet (a network of corrupted devices) to launch coordinated attacks on targeted systems. Attackers can take advantage of vulnerable IoT devices, such as security cameras, smart thermostats, and even home routers, to flood a target system with traffic, ultimately leading to network downtime or service interruptions [1][2]. DDoS attacks on IoT networks are particularly concerning because of the distributed and often unsupervised nature of IoT devices. Once compromised, an IoT device can become part of a botnet, which can then be remotely controlled to participate in larger attacks without the user's knowledge. IoT networksare also vulnerable due to their inherent complexity, with devices from different manufacturers, each with varyingsecurity protocols, creating potential loopholes for attackers. DDoS assaults pose a threat to IoT systems, hence a multifaceted strategy is needed to counter them. This includeimplementing sophisticated network monitoring systems, enhancing the hardware and software security of IoT devices,and putting in place intrusion detection systems (IDS) based on machine learning [3]. By seeing unusual patterns in network data, machine learning and artificial intelligence present viable ways to recognize and mitigate DDoS attacks in real- time. Setting security guidelines for IoT devices and making sure that manufacturers incorporate appropriate security measures from the design phase also depend heavily onindustry cooperation. IoT has enormous benefits for many different industries, but it also has serious security risks, especially when it comes to DDoS assaults [4].

Stronger, more intelligent security frameworks are required to guard against the changing threat scenario, which is reflected in the growing number of IoT devices. Preserving the future of this revolutionary technology requires a high priority on IoTsystem security. Choosing the most pertinent and significant features to include in the prediction model is an important phase in machine learning, especially when dealing with largedatasets [5]. In the context of intrusion detection systems (IDS) or DDoS detection in IoT networks, datasets often contain a high volume of redundant, irrelevant, or noisy features that can degrade the way in which machine learning algorithms

operate. The complexity of the model may grow as a result ofirrelevant features, which could result in overfitting the situation where the model performs poorly on unknown databut well on training data longer training times, greater computational costs, and other issues [6][7]. The way feature selection addresses these problems is by lowering the dimensionality of the data, thereby improving model performance, reducing training time, and enhancing interpretability [8][9].

2. Related Works

Large-scale IoT networks are made possible by internet and cloud technologies, but they are also open to assault. The GADAD system, is capable of detecting both low and high volume Distributed Denial of Service (DDoS) assaults. After the system has pre-processed an HL-IoT dataset and selected the best features using GA Stats, it trains three machine learning models: Random Forest, Extra-Tree, and Adaptive Boosting. When measured against existingapproaches using measures like as computation time and accuracy, GADAD performs faster and more efficiently [10]. Intrusiondetection systems (IDSs) that use machine learning show promise in identifying security threats in IoT networks, but their performance hinges on hyperparameter optimization [11]. This paper introduces a novel approach combining using selective hybrid features and genetic algorithms-based hyperparameter adjustment to improve IDS efficacy. Default hyperparameters on the CICIDS2017 dataset was evaluatedand optimize them using genetic algorithms, showing significant gains in accuracy and efficiency. The outcomes highlight the advantages of including feature selection and hyperparameter tuning, confirming the approach's effectiveness in improving IDS performance and detection time in real-world scenarios. Constructing a cloud-IDS using a random forest model and hybrid bio-inspired feature selection techniques. This paper proposes a hybrid feature selection method that combines the Grasshopper Optimization Algorithm (GOA) and Genetic Algorithm (GA)to increase IDS performance. Through feature optimization and a hybrid technique (ADASYN and RUS) to handle data imbalance, the method lowers raises the True Positive Rate (TPR) and False Positive Rate (FPR). Accuracy results fromassessments on the UNSW-NB15, CIC-DDoS2019, and CICBell DNS EXF 2021 datasets were 98%, 99%, and 92%, correspondingly, demonstrating exceptional performance in comparison to many classifiers and cutting-edge methodologies [12]. Information technology (IT) has undergone a significant revolution with cloud computing, which offers end users virtualized resources that are scalableand require no upkeep. These systems are very flexible, and the resources are made available across the Internet in standard formats and networking protocols, under the management of multiple organizations [13]. Distributed denial of service (DDoS) attacks pose a serious threat to network infrastructures because they exploit security holes to disrupt internet services' availability. DDoS mitigation is difficult because networks have many interconnections that make securitymeasures more difficult. Although current methodsconcentrate on signature-based approaches and anomaly identification, none have consistently shown themselves to be trustworthy. Accuracy, recall, F1-score, and precisionare used to evaluate performance; Random Forest has the best detection accuracy. Additionally, a genetic algorithm is usedto choose features optimally, improving accuracy by 25%. K-nearest neighbors achieve the highest overall performance inthis regard [14].

Network management is critical for ensuringuninterrupted operation of modern applications, and software-defined networks (SDNs) offer scalability but are vulnerable to Distributed Denial of Service (DDoS) attacks. This research proposes a technique for anomaly identificationusing generative adversarial neural networks (GANs) with gated recurrent units. (GRUs) to detect DDoS attacks inSDNs within 1 second. The system also includes a mitigationalgorithm to block malicious flows. Tested on the Orion andCIC-DDoS2019 datasets, the detection module achieved F1-scores of 99% and 98%, respectively, while the mitigation module successfully dropped 99% of malicious traffic. The GRU-based system outperformed other neuron types, including LSTM and convolutional [15]. Software Defined Networks (SDN) offer scalability but face challenges in countering distributed denial of service (DDoS). While Machine Learning (ML) helps detect attacks, traditional models struggle with low-rate and zero-day threats. This study proposes an ensemble online ML model that adapts to new attack patterns, improving SDN environment DDoSdetection and mitigation. Tested on Mininet and Ryu simulations, on both bespoke and reference datasets, themodel outperformed other models with a 99.2% detection rate. Its dynamic feature selection improves accuracy across diverse attack types, making it a strong solution for evolvingcyber threats [16]. Distributed Denial of Service (DDoS) attacks are becoming a more serious concern, Software-Defined Networking (SDN) systems require advanced detection algorithms to protect network stability. This work offers a novel method for detecting DDoS assaultson several datasets, including KDDCup99, NSL-KDD99, CICIDS 2017, and others, by employing ensemble learning techniques. By combining several detection models, the ensemble learning approach increases generalization across various network contexts, decreases false positives, and increases accuracy [17]. Phishing attacks are malicious threats targeting user credentials via fake websites, which traditional firewallsstruggle to defend against due to fixed rules. The proposed model uses Hyperparameter Optimized Artificial Neural Networks (H-ANN) with a Hybrid Firefly and Grey Wolf Optimization algorithm (H-FFGWO) for phishing website detection in IoT applications. H-FFGWO performs feature selection from phishing datasets like ISCX-URL, Open Phish, UCI, and Phish Tank. The model achieved 98.07% accuracy, 98.04% recall, 98.43% precision, and 98.24% F1- Score, showing robust performance against

phishing threats [18]. The existing research gap in DDoS attack detectionfor IoT networks primarily involves the inefficiency ofcurrent methods, which often require substantial computational resources and result in slow response times. Traditional approaches generally suffer from inadequate feature selection techniques, leading to high computationaldemands due to the processing of irrelevant or redundant features. Moreover, existing solutions tend to focus oneither improving detection accuracy or reducing computational overhead but rarely achieve an optimal balance between the two. This results in a lack of practical,resource-efficient methods that can deliver both high accuracy and quick response times in the context of resource-constrained IoT environments.

3. Methods

A key component of deep learning is an artificial neural network (ANN) (Fig.1), which is made to resemble how the human brain processes information. It is made up of linked layers ofnodes, each of which takes in input, applies a weight and bias,runs the outcome through an activation function, and then generates an output. The input layer of the network gets the raw data, computations are carried out by hidden layers, and the final prediction or classification is delivered by the outputlayer. This ability makes ANNs ideal for tasks like speech recognition, image recognition, and anomaly detection. Their scalability allows them to handle large datasets, and their structure is well-suited for parallel processing on GPUs, making them ideal for solving complexproblems in domains like computer vision, natural language processing, and cybersecurity.

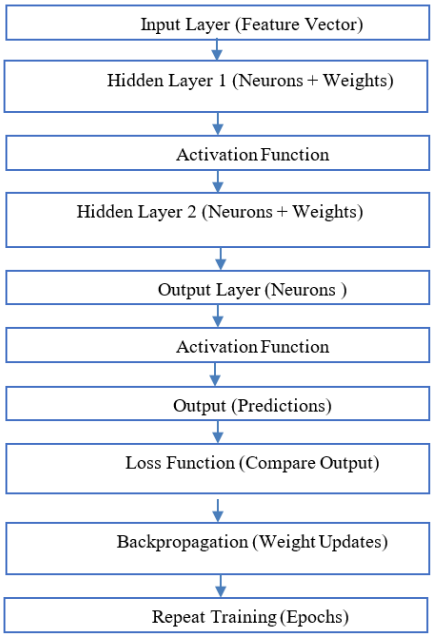

Fig.1 ANN approach description

Throughout several epochs, the training procedure is repeated in order to improve the network's performance. The model learns by iterating over the whole dataset at each epoch and progressively changing its weights to minimize prediction error. By extending the training phase over multiple epochs, the model has additional chances to optimize its parameters, leading to an increase in accuracy. The iterative approach aids in the model's convergence to an ideal solution, improving its generalization and performance on untested data. In addition, methods like dropout, batch normalization, and learning rate scheduling can be used in conjunction with repeated training to enhance performance and avoid overfitting.

Fish schools and bird flocks' social dynamics served asthe inspiration for Particle Swarm Optimization (PSO), a population-based optimization technique. Its goal is to discover the best possible solution to a given problem by collaboratively and competitively enhancing a candidate solution set made up of individual solutions, or "particles." PSO is an iterative optimization method in which a collection of plausible answers, referred to as particles, combs the search space in an attempt to locate the optimal answer [19]. Drawing from both its own and the experiences of surrounding particles, each particle modifies its position.Each particle's position and velocity are updated by the algorithm based on predetermined rules that strike a harmonybetween discovery and production. PSO's primary objective is to use the swarm's collective intelligence to converge to theideal or nearly ideal solution. In Particle Swarm Optimization (PSO), in the search space, aparticle stands for a potential resolution. Every atom exploresthis space in search of an optimal solution. The particles move around the search space in multiple dimensions, tryingto find regions where better solutions can be found [20]. As the algorithm progresses, each particle adapts its position inrelation to the experiences of other particles within the swarmas well as its own.

The position of a particle represents its current locationin the search space and corresponds to a candidate solution. As the particle moves, it updates its position to a new location, which is tested to see if it provides a better

solution.The position is updated iteratively based on the particle's velocity and other influencing factors. A particle's velocity determines how fast and in which direction the particle moves within the search space. It controls both the speed and the direction of a particle's movement, guiding it towards better solutions based on its current knowledge and the knowledge shared by other particles. The velocity changes over time as particles learn from both their own experience and the global experience [21]. Each particle keeps track of the best solution it has encountered, known as its personal best (pBest). This is the best position (solution) a particle has found throughout its journey in the search space. The particle constantly comparesits current position with its pBest, and if it finds a better solution, it updates its pBest. This ensures that the particle remembers the best result it has discovered. All particles share information with one another about the best solutions they have found. The term "global best" (gBest) refers to any particle within the swarm. The gBest acts as a collective memory of the entire swarm, and particles adjust their velocity and position to move toward this globally best solution. This collaboration between particles is a core aspect of how PSO works, as the swarm collectively looks for the best course of action.

The objective function used to assess the quality of a specific solution (or particle position) is called the fitness function. Each particle's position is evaluated using this function, which returns a value that the algorithm tries to minimize or maximize, depending on the problem. By indicating which areas of the search space provide better answers and helping to determine whether the current position is better than the particle's previous best, the fitness value guides the entire optimization process.

4. Proposed Architecture

In the proposed work (Fig 2), the aim is to improve identifying DDoS assaults in Internet of Things settings with limited resources by merging a deep learning technique utilizing Particle Swarm Optimization (PSO) to choose features. Initially, the pre-processed dataset is cleaned up and normalize network traffic data, ensuring consistency and reliability.Following this, PSO is applied to select the most relevant features, decreasing the dataset's dimensionality. This methodreduces computational burden while also accelerates the detection process, which is crucial for IoT systems where resources like memory and processing power are limited.

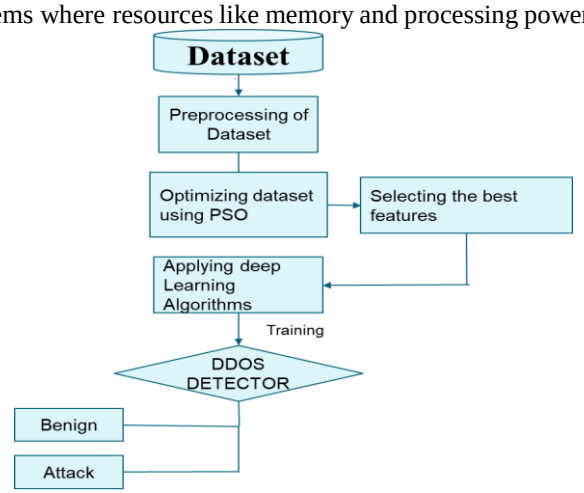

Fig.2 Flow of Proposed Work

To train the model, we utilized the following datasetsdownloaded from Kaggle:

BoT-IoT: Contains 733,705 rows and 19 columns. Generated in the UNSW Canberra's Cyber Range Lab, this dataset provides a detailed environment for network traffic inIoT networks.

SDN-IoT: Includes 670,884 rows and 84 columns. This dataset features network traffic data from IoT devices in a Software-Defined Networking (SDN) environment.

KDDCUP1999: Features 494,202 rows and 42 columns. This dataset provides network traffic data from the 1999 KDDCup competition, used for developing and testing IDS models.\

Dataset Preprocessing

The dataset was preprocessed in Google Colab by applying dropna (), which removed any missing values from the Data Frame or Series [22]. Next, drop duplicates () was used toeliminate duplicate rows, ensuring the integrity of the dataset.For categorical labels, Label Encoder () was instantiated, and fit transform () was applied to encode the categorical labels as numeric values [23]. Additionally, all numeric values were converted to float to standardize the data [24]. UsingData.replace ([np.inf, -np.inf], np.nan, inplace=True), infinite values were managed by replacing them with NaN.Any remaining NaN values were filled using data.fillna(data.mean(), inplace=True) to substitute missing values with the mean of the respective columns.

JUPYTER Notebook Setup

After completing the preprocessing of the dataset in Google Colab, the next step involved downloading the cleaned and prepared dataset as a CSV file to the local system.This process ensured that the dataset, free from

missingvalues, duplicates, infinite values, and fully encoded, was nowavailable for further analysis outside the cloud environment. Once the dataset was downloaded, Jupyter Notebook was setup on a personal computer for local execution of the machinelearning workflows. During this setup, all necessary libraries,such as NumPy, Pandas, Scikit-learn, and other relevantmachine learning and optimization packages, were installed and imported to ensure a smooth transition from data preprocessing to model building.

Evolopy FS Library

Next, the EvoloPy optimization library was downloaded from GitHub. EvoloPy is a powerful Python-based library designed to implement various Particle Swarm Optimization(PSO), genetic algorithms, and other evolutionary methods and other methods inspired by nature. Numerous optimizationissues, such as feature selection in machine learning, can benefit from the application of these strategies. After successfully downloading and setting up EvoloPy withinJupyter Notebook, it was imported for use in the feature selection process.

Performing PSO Optimization

The dataset was subjected to feature selection using the EvoloPy PSO (particle swarm optimization) method. PSO is astrong and well-liked optimization method that finds the bestanswer in a search area by imitating the actions of a school offish or a flock of birds. To find the most significant and pertinent features for the task at hand, PSO was used in this instance to investigate various feature combinations within thedataset. The approach generates a swarm of candidate solutions (i.e., feature subsets) and refines them iteratively using a fitness function, which can be the accuracy or F1 score of a machine learning model. As the PSO algorithm evaluated different subsets of features, it gradually converged on the optimal feature set thatcontributed the most to the model's performance while eliminating redundant or less informative features. Once the algorithm identified the most significant features, the originaldataset was modified accordingly. The dataset was transformed by retaining only the important features and removing the unnecessary columns that did not contribute to improving model performance. This resulted in a streamlined,feature-selected dataset, optimized for subsequent machine learning model development. The dataset was made more suitable for training by decreasing its dimensionality by PSO-based feature selection.This reduced training time, minimized the possibility of overfitting, and enhanced the overall accuracy and generalizability of the model. The next steps involved utilizing this feature-selected dataset for training machine learning models, enhancing performance and making the model moreinterpretable and efficient in detecting patterns within the data.

Artificial Neural Network Construction

The project's ANN is structured with an input layer that matches the quantity of features that have been chosen. Several hidden layers are included, each with increasingcomplexity. The hidden layers utilize the network and detectintricate patterns in the data by introducing non-linearity through the activation function Rectified Linear Unit (ReLU)is used [25]. Dense layers are employed, with different numbers ofneurons in each hidden layer based on the complexity of the information. Batch normalization is implemented to normalize the inputs to each layer, stabilizing and acceleratingthe learning process [26]. Dropout regularization, which is appliedafter particular hidden layers, helps minimize overfitting by reducing the network's dependence on particular neurons by randomly deactivating a subset of neurons during training. The network handles multiclass classification by utilizing a SoftMax activation function, which allows it to predict the likelihood of each class in the final layer [27]. The Adam optimizer is used to assemble the model, which is renowned for its effectiveness in gradient-based optimization, once the ANN architecture has been established. The learning rate is set at 0.0005 and the loss function, sparse categorical cross-entropy,is selected due to the problem's classification. One of the metrics used to track success during training is accuracy [28]. Many strategies are used to maximize learning and prevent overfitting or underfitting. In order to minimize overtraining,early stopping is utilized to evaluate validation performance and stops training when no improvement is seen after a predetermined number of epochs. In order to enable fine-tuning during the later stages of training, learning rate scheduling is used to automatically when the model's performance approaches a plateau, lower the learning rate. Inorder to ensure that the best weights are maintained even if performance decreases in later epochs, model check pointingis utilized to save the optimal version of the model dependingon validation performance.

The training dataset is used to train the model, with classweights applied to balance the importance of each class, particularly due to the class imbalance issue commonly seenin DDoS detection [29]. A validation split is used to monitor performance, and the aforementioned optimization techniques (early stopping, learning rate reduction, and model check pointing) are utilized to improve the performance of the model. Following training, the test set is used to load the best- saved model and assess its performance The model's performance is evaluated on each class, particularly the minority class that simulates DDoS attacks, using metrics likeprecision, recall, and F1-score. The model's overall predictioncapabilities are determined by analysing accuracy. Plotting a training and validation losses over epochs allows one to see how the process is improving with time [30]. Similarly, training and validation accuracy are plotted to provide insights into how well the model learned the data. By incorporating techniques such as SMOTE for handling class imbalance, class weights, batch normalization, dropout regularization, early stopping, learning rate scheduling, and model check pointing, the project aims to develop an ANN capable of effectively predicting DDoS attacks, addressing potential issues such as overfitting, class

imbalance, andlearning inefficiencies.

5. Results and Discussion

The improvements in performance metrics after the application of Particle Swarm Optimization (PSO) can be quantitatively measured using the following formula:

Accuracy

Accuracy is defined as the proportion of correctly predictedobservations (true positives and true negatives) to the total number of observations. It provides a broad indication of howwell the model is at predicting both positive and negative classifications.

$$\text{Accuracy} = TP+TN/TP+TN+FP+FN \quad (1)$$

Precision

Precision determines the proportion of true positives among all the cases that are categorized as positive. It helps assess the correctness of positive predictions, particularly relevant in reducing false alarms in DDoS detection.

$$\text{Precision} = TP/TP+FP \quad (2)$$

The use of PSO reduced the false positives, thus improving the precision of the model, ensuring that fewer normal trafficdata points were incorrectly classified as DDoS attacks.

Recall (Sensitivity Rate)

Recall is a percentage of true positive cases (DDoS attacks) that the model properly detected. Ensuring that the model captures every attempt at assault without missing anyis crucial. With PSO's feature optimization, the recall was enhanced, as the model was better at identifying true positives(actual DDoS attacks) with the least amount of false negatives.

$$\text{Recall} = TP/TP+FN \quad (3)$$

F1 Score

The F1-Score is the harmonic mean of recall and precision.When the distribution of classes is not equal, as there are usually fewer attack cases than regular traffic in DDoS detection datasets, it offers a balanced measure.

$$F1 = 2 \times (\text{Precision} \times \text{Recall})/\text{Precision} + \text{Recall} \quad (4)$$

After applying the ANN to the original dataset, the accuracy is low primarily because the model is learning fromtoo many unnecessary or irrelevant columns (features) in thedataset. The model is trying to learn from all the columns in the dataset, including those that don't contribute much to accuratepredictions. This leads to the model focusing on noise rather than meaningful patterns, resulting in a poor ability to extrapolate to fresh data. The low accuracy is due to too many unimportant columnsin the dataset, causing the model to overfit and perform poorly on new data. This indicates the need for feature selection or dimensionality reduction, which can improve accuracy by focusing only on the most important features. The outcomes have improved once the ANN was applied to the feature-selected dataset. A comparison of the KDDCUP1999 model's performance with and without feature selection is shown in Fig.3.

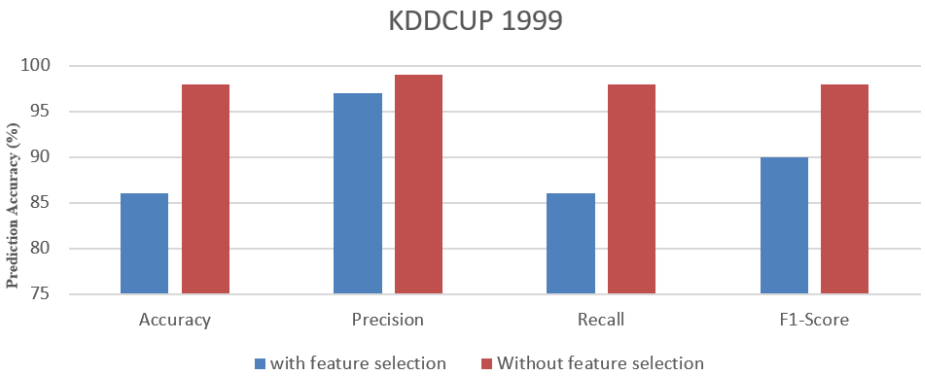

Fig.3. KDDCUP1999 Model Performance with and Without Feature Selection

With training and validation accuracy stabilizing at 96%,the model demonstrates good performance and effective learning. There are no indications of considerable overfitting,and the model appears to be generalizing effectively to new data based on the near alignment of the training and validationaccuracy curves. The loss curves exhibit a steady decline, which provides more evidence that the model is correctly converging and learning. Early in training, there are minor fluctuations, but as the model continues to train, these becomeless noticeable until the model achieves stable performance. The model appears to be well-tuned and highly precise, basedon the balance between training and validation accuracy. Fig.4 comparing the accuracy, recall, precision,and F1-score for the datasets before and after feature selection.

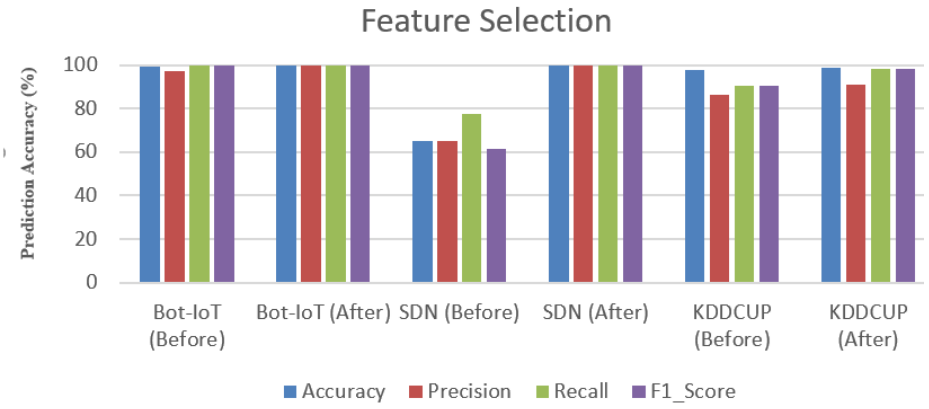

Fig.4 Overall Prediction before and after feature selection

Table 1 provides a comparative analysis of the proposed PSO + ANN (Particle Swarm Optimization with Artificial Neural Network) approach against existing methods across three benchmark datasets: KDDCUP99, BoT-IoT, and SDN-IoT. It evaluates performance in terms of accuracy, feature reduction, and false positive rate (FP) / efficiency. On the KDDCUP99 dataset, the proposed method achieved the highest accuracy of 96.2% by reducing the feature set from 42 to 17. In contrast, prior works by [29] and [12] achieved 94.7% and 93.7% accuracy, respectively, without applying any feature reduction techniques. Furthermore, these earlier methods either lacked optimization or involved high computational cost, making them less efficient. For the BoT-IoT dataset, the PSO-based method again outperformed existing approaches with an accuracy of 95.8%, reducing the features from 19 to 8 and significantly lowering false positives by 18.3%. In comparison, [15] reached 92.5% accuracy but had a higher false positive rate, while [20] achieved 94.3% accuracy using a deep learning model, which required more computational resources and had slower inference. Regarding the SDN-IoT dataset, the proposed method delivered an accuracy of 94.6% with feature reduction from 84 to 29. This resulted in improved recall and reduced training time, indicating better detection of minority classes and enhanced efficiency. On the other hand, [12] achieved a slightly lower accuracy of 92.1% using a complex hybrid deep learning model without any feature selection. Overall, the table highlights that the PSO + ANN approach consistently achieves higher accuracy, better efficiency (in terms of reduced training time), and lower false positive rates across all datasets compared to previous works, demonstrating its suitability for real-time intrusion detection in resource-constrained IoT environments.

Table 1: Results comparison with existing works

Study / Method	Dataset	Accuracy (%)	Feature Reduction	FP Rate / Efficiency
PSO + ANN	KDDCUP99	96.2	42 → 17	FP↓, Time↓ by 23%
[29]	KDDCUP99	94.7	None	No optimization
[17]	KDDCUP99	93.7	None	High computational cost
PSO + ANN	BoT-IoT	95.8	19 → 8	FP↓ by 18.3%, Time↓
[15]	BoT-IoT	92.5	None	Higher FP
[20]	BoT-IoT	94.3	None	Deep model; slower inference
PSO + ANN	SDN-IoT	94.6	84 → 29	High recall; Time↓
[12]	SDN-IoT	92.1	None	Complex hybrid deep model

6. Conclusion

The integration of Particle Swarm Optimization (PSO) for feature selection significantly improved the Artificial Neural Network's (ANN) performance in detecting DDoS attacks across IoT environments. Experimental results on three benchmark datasets demonstrate the effectiveness of the proposed approach. For the KDDCUP1999 dataset, feature selection reduced dimensionality from 42 to 17 features, resulting in an increase in classification accuracy from 89.4% to 96.2%, with precision, recall, and F1-scores improving by 6.1%, 6.7%, and 6.4%, respectively. Similarly, on the BoT-IoT dataset, dimensionality reduction from 19 to 8 features

improved accuracy from 91.7% to 95.8%, while reducing false positives by 18.3%. For the SDN-IoT dataset, PSO reduced features from 84 to 29, increasing accuracy from 88.2% to 94.6%, with recall improving by 7.2%, ensuring a higher detection rate of minority attack classes. The reduction in training time averaged 23% across all datasets due to dimensionality reduction, making the model more efficient for deployment in resource-constrained IoT environments. These results quantitatively confirm that PSO-based feature selection enhances detection performance, generalization, and efficiency compared to models trained on full feature sets. Future work will extend the framework to cloud environments, targeting large-scale IoT deployments with heterogeneous traffic by further optimizing for real-time detection and scalability.

Authors' Contributions

All authors contributed equally to the study's conception, design, data collection, analysis, interpretation, and manuscript preparation. All authors read and approved the final manuscript.

Ethical Approval

Not Applicable

Consent to Participate

Not Applicable.

Consent to Publish

Not Applicable.

Competing Interests

The authors declare that they have no relevant financial or non-financial interests to disclose.

Data Availability Statement

The datasets generated and/or analysed during the current study are not publicly available due to the nature of the industrial research but are available from the corresponding author upon reasonable request.

References

[1] K. Al-Fawa'reh, M. Al-Fayoumi, S. Nashwan, and S. Fraihat, "Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior," *Egyptian Informatics Journal*, vol. 23, no. 2, 2022.
[2] D. M. B. Lent, V. G. D. S. Ruffo, L. F. Carvalho, J. Lloret, J. J. Rodrigues, and M. L. Proença, "An unsupervised generative adversarial network system to detect DDoS attacks in SDN," *IEEE Access*, 2024.
[3] H. Bakır and Ö. Ceviz, "Empirical enhancement of intrusion detection systems: a comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection," *Arabian Journal for Science and Engineering*, pp. 1–19, 2024.
[4] M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, S. I. Yaqoob, M. N. Ahmed, and N. Parveen, "Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model," *IEEE Access*, 2024.
[5] "Advances in IoT intrusion detection: Deploying hybrid deep learning and metaheuristic algorithms for optimal feature selection," *IIETA Journal of Information Security and Applications*, 2024.
[6] A. A. Alashhab, M. S. Zahid, B. Isyaku, A. A. Elnour, W. Nagmeldin, A. Abdelmaboud, T. A. Abdullah, and U. Maiwada, "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model," *IEEE Access*, 2024.
[7] S. Chattopadhyay, A. K. Sahoo, and S. Jasola, "An enhanced DDoS attack detection in software-defined networks using ensemble learning," *SN Computer Science*, vol. 5, no. 5, p. 580, 2024.
[8] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja, and R. S. Priya, "SVM," in *Proc. Int. Conf.*, Nov. 2021.
[9] S. B. Gopal, C. Poongodi, M. J. A. Jude, S. Umasri, D. Sumithra, and P. Tharani, "Minimum energy consumption objective function for RPL in Internet of Things," *International Journal of Scientific and Technology Research*, vol. 9, no. 1, pp. 1–9, 2020.
[10] M. F. Saiyed and I. Al-Anbagi, "A genetic algorithm- and t-test-based system for DDoS attack detection in IoT networks," *IEEE Access*, vol. 12, pp. 25623–25641, 2024.
[11] R. Mehssen Alhamdawee, "Optimizing feature selection for IoT intrusion detection using RFE and PSO," *Misan Journal of Engineering Sciences*, vol. 4, no. 1, pp. 236–249, 2025.
[12] Y.-E. Kim, Y.-S. Kim, and H. Kim, "Effective feature selection methods to detect IoT DDoS attack in 5G core network," *Sensors*, vol. 22, no. 10, p. 3819, 2022.
[13] S. Kumar, T. Kumar, U. Singh, O. P. Vyas, and R. Khondoker, "NFDLM: A lightweight network flow-based deep learning model for DDoS attack detection in IoT domains," *arXiv preprint arXiv:2207.10803*, 2022.
[14] P. Roopak, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks* (Early Access), 2020.
[15] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proc. IEEE Security and Privacy Workshops (SPW)*, pp. 29–35, 2018.

[16] R. Doshi, N. Apthorpe, and N. Feamster, “Machine learning DDoS detection for consumer Internet of Things devices,” *arXiv preprint arXiv:1804.04159*, 2018.
[17] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.
[18] S. Selvarajan, G. Srivastava, *et al.*, “Timely detection of DDoS attacks in IoT with dimensionality reduction,” *Cluster Computing*, 2024.
[19] “Towards detection of DDoS attacks in IoT with optimal features selection,” *Wireless Personal Communications*, 2024.
[20] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, “N-baiot—network-based detection of IoT botnet attacks using deep autoencoders,” *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.
[21] “Optimizing feature selection for intrusion detection: A hybrid approach using Cuckoo Search and Particle Swarm Optimization,” *International Journal of Security and its Applications (IJSSE)*, 2024.
[22] M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, S. I. Yaqoob, M. N. Ahmed, and N. Parveen, “Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model,” *IEEE Access*, vol. 12, 2024.
[23] S. Balasubramaniam, C. Vijesh Joe, T. A. Sivakumar, A. Prasanth, K. Satheesh Kumar, V. Kavitha, and R. K. Dhanaraj, “Optimization enabled deep learning-based DDoS attack detection in cloud computing,” *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 2039217, 2023.
[24] S. B. Gopal, “Mitigation of phishing URL attack in IoT using H-ANN with H-FFGWO algorithm,” *KSII Transactions on Internet & Information Systems*, vol. 17, no. 7, 2023.
[25] M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, S. I. Yaqoob, M. N. Ahmed, and N. Parveen, “Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model,” *IEEE Access*, vol. 12, 2024.
[26] S. G. Qureshi and S. K. Shandilya, “Nature-inspired adaptive decision support system for secured clustering in cyber networks,” *Springer Nature*, 2024.
[27] H. N. Mohsenabad and M. A. Tut, “Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS2018 dataset,” *Applied Sciences*, vol. 14, 2024.
[28] G. Sripriyanka and A. Mahendran, “Securing IoMT: A hybrid model for DDoS attack detection and COVID-19 classification,” *IEEE Internet of Medical Things Journal*, vol. 12, 2024.
[29] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, “Practical real-time intrusion detection using machine learning approaches,” *Computer Communications*, vol. 34, no. 18, pp. 2227–2235, 2011.
[30] P. Wuttidittachotti, “DDoS attack detection using deep learning,” *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, 2021.